

# Proiectarea filtrelor IIR prin metoda Pade

Laborator 2, PSS

## Table of contents

<b>1</b>	<b>Obiectiv</b>	<b>1</b>
<b>2</b>	<b>Noțiuni teoretice</b>	<b>1</b>
2.1	Metoda Pade . . . . .	2
2.2	Algorithm matematic . . . . .	2
<b>3</b>	<b>Noțiuni Matlab</b>	<b>3</b>
3.1	Rezolvare sistem de ecuații cu <code>linsolve()</code> . . . . .	3
3.2	Calcularea răspunsului la impuls cu <code>impz()</code> . . . . .	3
<b>4</b>	<b>Exerciții teoretice</b>	<b>4</b>
<b>5</b>	<b>Exerciții practice</b>	<b>4</b>
<b>6</b>	<b>Întrebări finale</b>	<b>5</b>

## 1 Obiectiv

Proiectarea filtrelor IIR prin metoda Pade.

## 2 Noțiuni teoretice

Proiectarea unui filtru înseamnă găsirea valorilor coeficienților de la numărătorul și numitorul unei funcții de sistem,  $b_0, b_1, \dots, b_M$  și  $a_1, a_2, \dots, a_N$ , astfel încât filtrul să rezulte cu caracteristicile dorite.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + b_N z^{-N}}$$

## 2.1 Metoda Pade

În metoda Pade, dorim să găsim un sistem  $H(z)$  a cărui răspuns la impuls  $h[n]$  să fie similar cu un răspuns la impuls dorit  $h_d[n]$ . Răspunsul la impuls dorit  $h_d[n]$  poate fi, de exemplu, răspunsul la impuls al unui filtru ideal trece-jos, trece-bandă sau trece-sus.

Ideea metodei Pade este următoarea: se găsesc  $b_0, b_1, \dots, b_M$  și  $a_1, a_2, \dots, a_N$  astfel încât **primele eșantioane** ale  $h[n]$  să fie **identice** cu cele ale  $h_d[n]$ .

## 2.2 Algorithm matematic

1. Se pornește de la ecuația cu diferențe generală a sistemului:

$$y[n] = -a_1 y[n-1] - \dots - a_N y[n-N] + b_0 x[n] + b_1 x[n-1] + \dots + b_M x[n-M]$$

2. Dacă intrarea este  $\delta[n]$ , ieșirea va fi  $h[n]$ , și prin urmare:

$$h[n] = -a_1 h[n-1] - \dots - a_N h[n-N] + b_0 \delta[n] + b_1 \delta[n-1] + \dots + b_M \delta[n-M]$$

3. Egalăm primele  $M + N + 1$  eșantioane cu cele ale  $h_d[n]$ . Numărul de eșantioane este același cu numărul total de coeficienți care trebuie obținuți. Se presupun condițiile inițiale  $h[-1] = h[-2] = \dots = 0$  (dacă nu se specifică altfel).

Se obține următorul sistem:

$$\begin{aligned} h[0] &= b_0 & &= h_d[0] \\ h[1] &= -a_1 h_d[0] + b_1 & &= h_d[1] \\ &\dots & & \\ h[M] &= -a_1 h_d[M-1] - \dots - a_M h_d[0] + b_M & &= h_d[M] \\ h[M+1] &= -a_1 h_d[M] - \dots - a_M h_d[M+1-N] & &= h_d[M+1] \\ &\dots & & \\ h[M+N] &= -a_1 h_d[M+N-1] - \dots - a_M h_d[N] & &= h_d[M+N] \end{aligned}$$

Rezultă un sistem cu  $M + N + 1$  ecuații, suficient pentru a găsi un număr total  $M + N + 1$  de necunoscute ( $a_i$  și  $b_i$ ).

4. Se rezolvă sistemul. La rezolvarea manuală, putem folosi următoarea observație:

- Ultimele  $N$  ecuații nu depind de  $b_i$ , doar de  $a_i$ . Acestea se rezolvă ca un sistem separat, mai mic, din care se obțin valorile  $a_i$ .
- Apoi trecem la primele  $M + 1$  ecuații. Se înlocuiesc  $a_i$  cu valorile găsite mai sus și se obțin  $b_i$ . Fiecare ecuație furnizează o valoare pentru un  $b_i$ .

## 3 Noțiuni Matlab

### 3.1 Rezolvare sistem de ecuații cu `linsolve()`

Pentru rezolvarea unui sistem de ecuații în Matlab, se poate folosi funcția `linsolve()`.

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 5 \\ 7 & 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

Soluție Matlab:

```
% Construct left vector and the system matrix
y = [1; 1; 1];
A = [1 2 3; 4 5 5; 7 8 9];

% Solve the system
x = linsolve(A, y);
```

Rezultat:

```
x =

-1.0000
 1.0000
 0.0000
```

### 3.2 Calcularea răspunsului la impuls cu `impz()`

Se folosește funcția `impz(b, a, n)`, unde:

- **b** = vector cu coeficienții de la numărător
- **a** = vector cu coeficienții de la numitor
- **n** = numărul de eșantioane din  $h[n]$  dorite

$$H(z) = \frac{2 + 0.3z^{-1} - 0.4z^{-2}}{1 - 0.2z^{-1} + 0.1z^{-2}}$$

```
% Construct the coefficient vectors
b = [2, 0.3, -0.4];
a = [1, -0.2, 0.1];

% Find first 20 samples of the impulse response
h = impz(b, a, n);
```

## 4 Exerciții teoretice

1. Să se determine prin metoda Pade parametrii filtrului cu funcția de sistem

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}},$$

dacă răspunsul la impuls dorit este

$$h_d[n] = \left(\frac{1}{3}\right)^n \cos(n\pi) u[n] + u[n-3]$$

## 5 Exerciții practice

1. Rezolvați numeric în Matlab sistemul de ecuații aferent metodei Pade de la exercițiul anterior, folosind funcția `linsolve()`.

$$\begin{bmatrix} h_d[0] \\ h_d[1] \\ h_d[2] \\ h_d[3] \\ h_d[4] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -h_d[0] & 0 & 0 & 1 & 0 \\ -h_d[1] & -h_d[0] & 0 & 0 & 1 \\ -h_d[2] & -h_d[1] & 0 & 0 & 0 \\ -h_d[3] & -h_d[2] & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

2. Calculați cu funcția `impz()` primele 10 eșantioane ale răspunsului la impuls pentru filtrul proiectat mai sus.

Verificați că primele 5 eșantioane sunt egale cu cele din  $h_d[n]$ . Dar celelalte?

3. Implementați în Matlab o funcție generală care să implementeze metoda Pade pentru orice ordin și orice  $h_d[n]$ :

```
function [b,a] = metodapade(ordin, hd)
...
end
```

Funcția va primi ca argumente:

- **ordin**: ordinul filtrului dorit
- **hd**: un vector cu răspunsul la impuls dorit (primele eșantioane)

Funcția va returna coeficienții funcției de sistem a filtrului proiectat:

- **b**: coeficienții de la numărător
- **a**: coeficienții de la numitor

4. Verificați cu această funcție rezultatul obținut la exercițiul 1
5. Utilizați funcția `impz()` pentru a găsi răspunsul la impuls al unui sistem cu

$$H(z) = \frac{1 - 1.7z^{-1} + 0.7z^{-2}}{1 + 1.3z^{-1} + 0.4z^{-2}}.$$

Apoi folosiți funcția `metodapade()` pentru a aproxima un filtru de ordin 2 din acest răspuns la impuls. Se obțin coeficienții filtrului original?

Dar dacă aproximăm un filtru de ordin 3?

6. Să se încarce un semnal audio în Matlab și să se filtreze cu filtrul proiectat mai sus. Redați semnalul filtrat la ieșirea audio a sistemului.

## 6 Întrebări finale

1. TBD